

OpenDaylight SDN and NFV Integration in OpenStack Cloud: OpenSource Approach for Improving Network Services



Hicham Boudlal, Mohammed Serrhini, and Ahmed Tahiri

Abstract Introducing software-defined networking and network function virtualization has brought new opportunities in cloud, enabling dynamic and autonomous configuration as well as provisioning of resources in the cloud. In this article, we highlight how OpenDaylight can be integrated with OpenStack to enhance network services. After describing SDN with NFV and how they compromise cloud services, we present some advantages of SDN in cloud networks. In addition, the deployment of an SDN solution on a high-performing cluster is presented. Finally, this cluster is integrated into the cloud OpenStack to enrich its services.

Keywords Cloud computing · SDN · Network architecture · NFV · OpenDaylight · OpenStack · OpenFlow

1 Introduction

Cloud computing has taken the IT industry and service providers into a whole different era, redefining the way IT resources and services are provisioned and used. In cloud computing, separate and distributed hardware resources, such as storage and computing capacity, can be acquired and utilized on demand, providing low-cost scalability and elasticity for applications. Traditional cloud networks typically had combined control and data plane where router device decides what path to take the traffic through policy and routing algorithm.

Software-defined networking (SDN) and network functions virtualization (NFV) hold the key to making a significant impact in this area. SDN relies on separating network control of data forwarding functionality, which allows controllers to directly program and abstract the underlying infrastructure as an abstraction of the network's top-level service functionality and network applications. NFV provides an alternative approach to designing, deploying, and managing network resources. Separating

H. Boudlal (✉) · M. Serrhini · A. Tahiri
Faculty of Sciences, Mohammed First University, BV Mohamed VI, BP 717, 60000 Oujda,
Morocco
e-mail: hicham.boudlal5@gmail.com

the functions of the network, like firewalls, threat identification, etc., from owned hardware devices to allow them to be used more efficiently. Proprietary hardware devices are used in software and are deployed where needed [1–4].

The structure of the paper is as follows: An overview of cloud computing is presented in Sect. 2. A network architecture enabling SDN is presented in Sect. 3. Following, benefits of SDN in the cloud are described in Sect. 4. We illustrate our proposed architecture in Sect. 5, the implementation of SDN and NFV in the OpenStack cloud and discussion. Conclusion and future work are outlined in Sect. 6.

2 Overview of the Cloud

For several years, a concept called cloud computing has been researched and practiced and has been classified into three categories of services: Platform as a Service (PaaS), Software as a Service (SaaS), and Infrastructure as a Service (IaaS). PaaS is based on the sub-layer, where developers of applications are able to use services of the platform delivered through the cloud. While SaaS delivers full-featured software to cloud clients, integrating cloud-based messaging services, networking services, scheduling services, and essentially all of the programs that run on cloud. IaaS is its most basic, providing virtual machine servers as well as the associated infrastructure to clients in the cloud. The infrastructure is used for any purpose by the buyers, for example to deploy their owned servers or to develop applications that run on the VMs. Both SaaS and PaaS are deployed when using IaaS. However, for cloud services to be delivered, this requires the provider to build and support a very large-scale cloud data center(s), where thousands or physical hosting devices are interconnected over thousands of switches in the network. Given the considerable complication of large-scale network connectivity, this requires a provider to look at the network in the data center through a different angle compared to the standard network [5–7].

3 SDN Enabled the Architecture of the Network

Software-defined networking (SDN) is an evolutionary concept that has changed the architecture of traditional networks. They actually separate the data plane of the control plane and place that in a central server called a controller. This removes network management complexity by placing it in the hands of a software controller, while providing abstracted control of the subjacent infrastructure. Both the data plane and the simple hardware components of the network are enabled, making it immediately programmable as well as being managed in a centralized environment [8]. Figure 1 shows the differences in concept of a traditional network and an SDN network. In the traditional model, a control plane is positioned above the routing plane within every device in a network, with discrete decision made by every commutator processing collected data from neighboring devices on a highly decentralized scale.

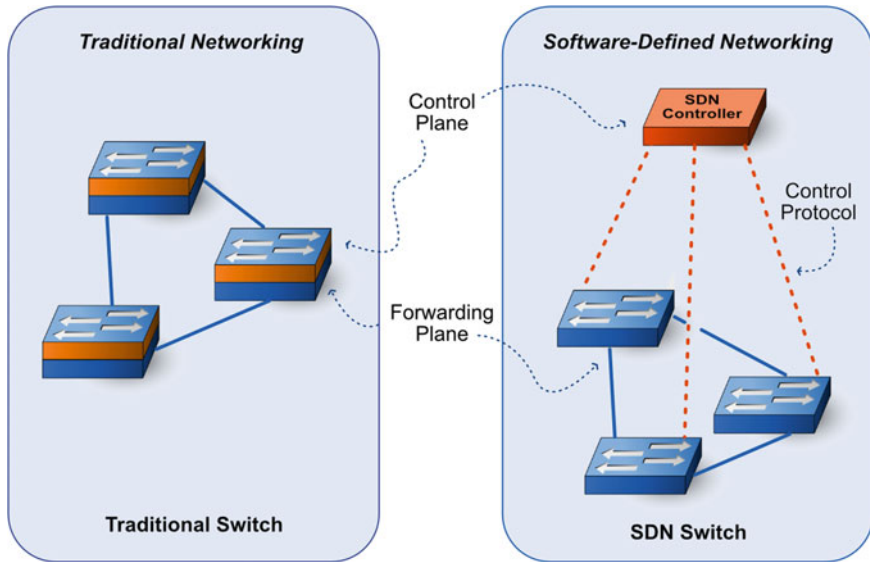


Fig. 1 Traditional networks compared to SDN

However, SDN puts the control plane in a centralized software controller that can monitor the network globally. Using information collected centrally, the controller is able to get simpler choices and deploy them to the transfer devices dynamically and efficiently [5, 9].

Open Networking Foundation (ONF) [10] proposed an SDN reference model as shown in Fig. 2 which is composed of three layers, that is, infrastructure layer, control layer, and application layer, stacked on top of each other.

The application layer includes programs communicating with a controller and requesting necessary resources. The control layer supplies logic-centric control functionality that monitors network behavior via opened interfaces. Typically, this is accomplished through a software component which can translate application requirements to data plane rules and provide the network information to them. The infrastructure layer is composed of the various components and low-level network devices involved in switching and transferring packets.

For interacting among different layers, there are two interfaces employed: the northbound interface, which assures that the applications communicate with a controller. The southbound interface ensures that the data path network elements communicate directly with the control layer, enabling the transfer rules to be controlled programmatically. The most extensive protocol used in this interface is OpenFlow [11].

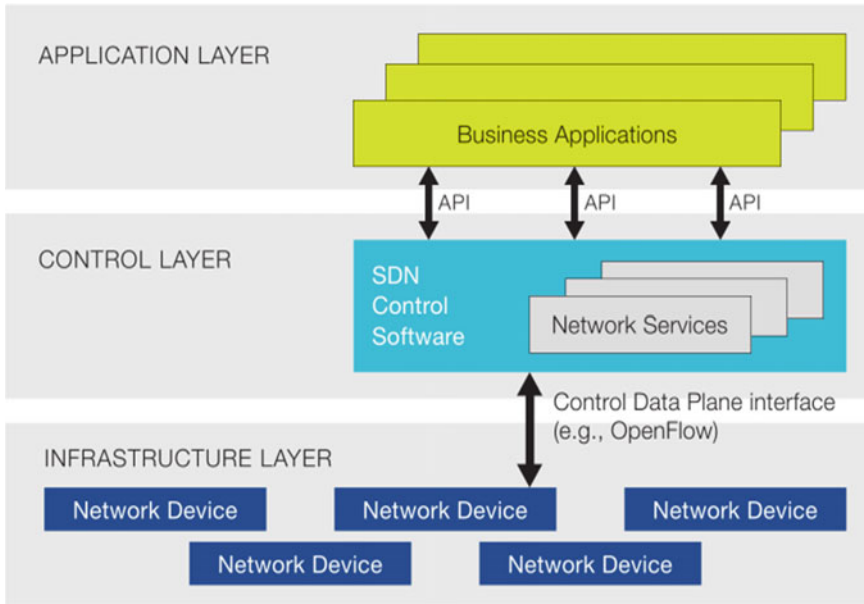


Fig. 2 SDN architecture

4 SDN Advantages in Cloud

Software-defined networking is the latest revolution in networking innovations [12], and cloud computing changes the way people have been doing computing and business [13]. Cloud networks are highly exposed and robust in providing services; therefore, while we could utilize SDN on a traditional network, there are multiple benefits to doing so.

Cloud-based centralized network provisioning: Not only can SDN along with programmability as well as virtualized technologies offer design flexibility, provisioning, and predefined scheduling, it provides the capacity for dynamic scalability to meet runtime requirements.

Integrated approach to business administration: With SDN, experimenting is possible inside virtualized network topologies and introducing novel alternatives is also streamlined and economical.

Intelligent security: With SDN, an easy and agile central application interface is provided for managing QoS policies, security, and controlling attacks.

Cloud abstraction: With SDN, network abstractions are essentially created to enable more agility and app-responsive behavior among a variety of cloud-IoT appliances. The network components comprising massive data center platforms are all manageable from the SDN controller.

Reduced downtime: SDN also supports the creation of configuration snapshots, allowing for quick recovery from upgrade failures.

Guaranteed content delivery: The capability to control and shape data traffic is a key benefit of software-defined networking. High-quality video streaming becomes easier as SDN provides improved network responsiveness to deliver a seamless user experience.

5 Proposed Architecture, Implementation, and Discussion

5.1 Proposed Architecture

An illustration of our proposed architecture is shown in Fig. 3 as an approach for improving network services, and the machine descriptions are as Table 1.

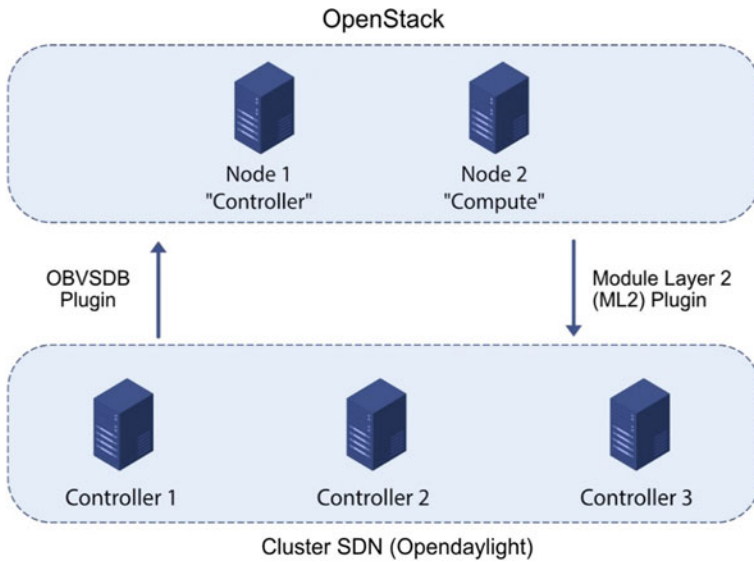


Fig. 3 Integration OpenStack and OpenDaylight

Table 1 Machine descriptions

Number of machine	Operating system	H/W detail	Purpose
1	Centos7	500 GB HDD 4 GB DR3 memory	Node 1 “Control”
1	Centos7	500 GB HDD 4 GB DR3 memory	Node 2 “Compute”
1	Ubuntu 14.04 X32	500 GB HDD 4 GB DR3 memory	Controller 1
1	Ubuntu 14.04 X32	500 GB HDD 4 GB DR3 memory	Controller 2
1	Ubuntu 14.04 X32	500 GB HDD 4 GB DR3 memory	Controller 3

5.2 Deploying an SDN Solution on a High-Performing Cluster

To ensure that the controller is more reliably available and to achieve the goal of high availability, we have opted to use OpenDaylight’s clustering technology. The installation procedure and the easy step-by-step installation are available at [14].

After running and configuring the three controllers in a cluster, we tested the cluster using a cluster monitoring tool. It will monitor the status of each node in the cluster, displaying two states—leader and follower. When a node is not working, it displays an error message. The leader state signifies that the node is in charge of this or that fragment, whereas follower is a safeguard state for a fragment that is under another node’s supervision.

Figure 4 shows the fragment for which the controller is responsible. The fragment names are displayed along X-direction, whereas controller roles appear along Y-direction. Follower and leader are displayed in the center of the window as shown in the figure. In this way, the status of the cluster can be monitored.

```

<3      default  toaster  inventory  topology
member-2 Leader    Follower Leader    Follower
member-3 Follower Leader    Follower Leader
member-1 Follower Follower Follower Follower
Press q to quit.

```

Fig. 4 Monitoring tool for clusters

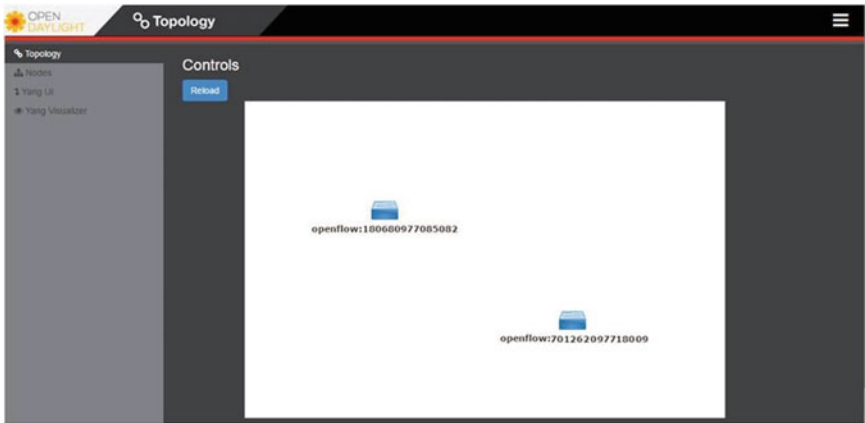


Fig. 5 DLUX showing Open vSwitch nodes

5.3 *OpenDaylight SDN Cluster Integration with OpenStack Cloud Platforms*

The OpenStack [15] can use OpenDaylight [16] as a network management provider via the Modular Layer 2 (ML2) plugin to the north. OpenDaylight handles network flows to OpenStack compute nodes via the OVSDB plugin in the south [17].

After the successful installation of OpenStack in two nodes (controller and compute), the installation procedure and easy step-by-step installation can be found in [18], and then, we moved on to its integration with the OpenDaylight cluster.

To achieve the integration, two main steps had to be taken: enabling OpenStack to utilize OpenDaylight as a provider of network management via the ML2 northward plugin and then OpenDaylight for managing the network flows for OpenStack nodes via the OVSDB southward plugin. After these steps, we got the following result. The two Open vSwitch of both compute and control nodes appeared on OpenDaylight as shown in Fig. 5.

Now, we have just to create and launch the instances. At this point, we confirm that the OpenStack integration is complete and the virtual machine traffic is controlled by OpenDaylight.

5.4 *Discussion*

The proposed architecture includes SDN and NFV plugin integrated into the OpenStack cloud network. SDN provides various new applications like resiliency, load balancing, and quality of service for cloud services to provide users for better flexibility and enhanced cloud service. Migrates traditional cloud network services into

SDN offers both flexibility and reliability through a programmable controller named SDN controller. NFV with SDN brings robust control of network through network virtualization as well as programmable routing of the network.

6 Conclusion and Future Work

The introduction of SDN has brought new varieties of approaches to the virtualization and monitoring of networks both within and external the cloud. In addition to NFV, SDN has enhanced cloud network control and expedited the delivery of novel network services. As an example of the applicability of integrating cloud computing and SDN paradigms, we demonstrated a practical study of OpenStack and OpenDaylight deployment. OpenDaylight and OpenStack are both large open-source projects that are backed by end-user and industry sponsored communities. Through our research, we have realized this insight. We incorporated SDN into cloud computing. Finally, it will be interesting to evaluate the overall performance of the proposal.

References

1. Manzalini A, Crespi N (2015) SDN and NFV for network cloud computing: a universal operating system for SD infrastructures. In: 2015 IEEE fourth symposium on network cloud computing and applications (NCCA), pp 1–6, IEEE, Munich. <https://doi.org/10.1109/NCCA.2015.11>
2. Battula LR (2014) Network security function virtualization (NSFV) towards cloud computing with NFV over Openflow infrastructure: challenges and novel approaches. In: 2014 international conference on advances in computing, communications and informatics (ICACCI), pp 1622–1628, IEEE, Delhi, India <https://doi.org/10.1109/ICACCI.2014.6968453>.
3. Singh S, Jha RK (2017) A survey on software defined networking: architecture for next generation network. *J Netw Syst Manage* 25:321–374. <https://doi.org/10.1007/s10922-016-9393-9>
4. Yi B, Wang X, Li K, Sk D, Huang M (2018) A comprehensive survey of network function virtualization. *Comput Networks* 133:212–262. <https://doi.org/10.1016/j.comnet.2018.01.021>
5. Son J, Buyya R (2018) A taxonomy of software-defined networking (SDN)-enabled cloud computing. *ACM Comput Surv* 51:1–36. <https://doi.org/10.1145/3190617>
6. Velte AT, Velte TJ, Elsenpeter RC (2010) *Cloud computing a practical approach*. McGraw-Hill, New York
7. Zhang Q, Cheng L, Boutaba R (2010) Cloud computing: state-of-the-art and research challenges. *J Internet Serv Appl* 1:7–18. <https://doi.org/10.1007/s13174-010-0007-6>
8. Rowshanrad S, Namvarasl S, Abdi V, Hajizadeh M, Keshtgary M (2014) A survey on SDN, the future of networking. *JACST* 3:232. <https://doi.org/10.14419/jacst.v3i2.3754>
9. Mishra V (2017) A survey on service-oriented network virtualization software defined networking for cloud computing. 12:7
10. Open Networking Foundation, <https://opennetworking.org/>.
11. McKeown N, Anderson T, Balakrishnan H, Parulkar G, Peterson L, Rexford J, Shenker S, Turner J (2008) OpenFlow: enabling innovation in campus networks. *SIGCOMM Comput Commun Rev* 38:69–74. <https://doi.org/10.1145/1355734.1355746>

12. Jain R, Paul S (2013) Network virtualization and software defined networking for cloud computing: a survey. *IEEE Commun Mag* 51:24–31. <https://doi.org/10.1109/MCOM.2013.6658648>
13. Xia W, Wen Y, Foh CH, Niyato D, Xie H (2015) A Survey on software-defined networking. *IEEE Commun Surv Tutor* 17:27–51. <https://doi.org/10.1109/COMST.2014.2330903>
14. Setting Up Clustering—OpenDaylight Documentation Nitrogen documentation, <https://docs.opendaylight.org/en/stable-nitrogen/getting-started-guide/common-features/clustering.html>.
15. Open Source Cloud Computing Infrastructure, <https://www.openstack.org/>.
16. Platform Overview, <https://www.opendaylight.org/about/platform-overview>.
17. Open Daylight integration with OpenStack: a tutorial—Superuser, <https://superuser.openstack.org/articles/open-daylight-integration-with-openstack-a-tutorial/>.
18. Packstack—RDO, <https://www.rdoproject.org/install/packstack/>.